



TITLE:

xfy上のMathML content markupエディタの実装と評価 (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

河田, 貴幸; 片岡, 正彰; 甲斐, 博; 田村, 恭士

---

CITATION:

河田, 貴幸 ...[et al]. xfy上のMathML content markupエディタの実装と評価 (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2009, 1666: 164-171

ISSUE DATE:

2009-10

URL:

<http://hdl.handle.net/2433/141065>

RIGHT:

# xfy 上の MathML content markup エディタの実装と評価

河田 貴幸

TAKAYUKI KAWATA

愛媛大学大学院理工学研究科

EHIME UNIVERSITY

片岡 正彰

MASAAKI KATAOKA

愛媛大学大学院理工学研究科

EHIME UNIVERSITY

甲斐 博

HIROSHI KAI

愛媛大学大学院理工学研究科

EHIME UNIVERSITY

田村 恭士

YASUSHI TAMURA

株式会社ジャストシステム

JUSTSYSTEMS Co.

## Abstract

xfy はジャストシステムにより提案された XML の編集ソフトウェアの一つである。複数のボキャブラリを一つの文書に含むような複合文書の編集ができるという特徴がある。数学文書中の MathML の利用方法としては単なる表示の他、既存の文書の数式の再利用・再計算があげられる。数式の計算や再利用を行うには、数学的な意味を記述する MathML Content Markup (以下、Content Markup) を利用することが望ましい。我々は、Content Markup を編集するための技術を検討し、xfy 上の Content Markup 編集環境の実装を行っている。本研究では、Content Markup の編集方法とその実装の評価について検討する。

## 1 はじめに

xfy は、XHTML や SVG 等のあらゆる XML ボキャブラリが組み合わせられた複合 XML 文書をシームレスに扱うことを目的としてジャストシステム社で設計された統合 XML アプリケーション開発／実行プラットフォームである。また、そのユーザインタフェースに関する特徴として、XML 文書の作成・編集を WYSIWYG で行うことができる。

数学文書は一般に多くの XML ボキャブラリを利用して編集されると考えられる。例えば、文書は XHTML、図は SVG、数式は MathML などである。xfy は複合 XML 文書に対する編集環境なので、このように複数のボキャブラリを持つ数学文書の編集に適している。我々は、数学文書内でシームレスに数式の編集や計算や描画が行える編集システムの開発を行っている。

MathML には、数式の表現のみを表す Presentation Markup と、数式の意味を表現する Content Markup がある。文書内で数式の計算も行う場合には意味を表現する Content Markup が適している。そこで、本研究では、xfy 上で Content Markup を WYSIWYG で編集するプラグインの実装を検討する。

## 2 Content Markup の入力インタフェース

本節では Content Markup プラグインが備える入力インタフェースについて述べる。Content Markup の編集を容易にするためには、GUI による入力インタフェースを備える必要がある。Padovani らは数式エ

ディタが備えるべき入力インタフェースを既存のエディタの様々な挙動を調査し、望まれる機能 [1] を次のようにまとめている。

**入力スロットの表示** 入力スロット (以下, スロット) は, ユーザに対してデータが未入力である部分を明示的に表示することで, その位置に値の入力を促すためのものである。入力スロットの表示は “?” や “■” 等のように表現は様々である。

**編集位置の表示とその移動** 数や変数等を挿入する位置を示すためにキャレットが用いられる。一般にこのキャレットは点滅する垂直の棒として表現される。また, キャレットは編集位置を表すものであるため, 移動可能である必要がある。そして, その望ましい挙動として方向キーによる上下左右への幾何的な移動ができ, かつ, その移動が対称であることが望まれる。対称な動きとは右方向キーによる移動後, 左方向キーにより以前の移動経路と逆順をたどることができる動きであり, 上下移動についても同様である。

**編集領域の選択** 現在選択されている領域をフォーカス領域と呼び, その領域が編集可能であることを表す。フォーカス領域には入力スロットを含む。またフォーカス領域の表示には, ある領域を実線や破線の矩形で囲んだり, あるいは領域内の文字の色や背景色を反転させる等の手法が用いられる。フォーカス領域の選択はマウスによるポインティングやドラッグ操作, あるいは Shift キー+方向キーによるキャレットの移動等で行われる。

**カット (コピー) & ペースト** フォーカス領域を切り取り, あるいはコピーして別の編集位置に貼り付ける編集操作である。この編集操作により, 既存の数式の記述情報を再利用することが容易となる。

**テンプレートとオーバーレイ** Content Markup において数式の持つ構造を入力する方法としてテンプレートやオーバーレイが利用できる。これらの入力方法は, 数式の構造を表したアイコン等を選択することでキャレット位置に入力できる。テンプレートとオーバーレイは類似する入力方法であるが, 前者の方法は先に数式の構造を与えた後に残りの入力スロットに対して数式を入力する方法である。一方, 後者の方法では既存の数式に対して新たに別の数式の構造を与えることができる入力方法である。

**数式の直接編集** 数式を容易に編集するためには, テキストエディタでのキーボード入力による直接編集が望ましい。多項式のような頻繁に利用される数式を直接編集できることは, 編集時におけるキーボードとマウスの交互利用回数の軽減に繋がるため, ユーザの編集に対する煩わしさをも軽減できると考えられる。

Content Markup プラグインではこれらの入力インタフェースを備えるべく実装を行っており, 現状での対応状況を表 1 に示す。以下では各指標に関する実装について述べる。

**入力スロット** Content Markup プラグインにおける入力スロットの表記は “[?]” であり, `math` 要素が子要素を持たない場合やトークン要素が値を持たない場合等に表示される。図 1 は実際に Content Markup プラグイン上で表示される入力スロットの様子である。

**キャレットの表示と幾何的な移動** Content Markup プラグインにおけるキャレットは青色の縦棒として表現している。但し, 現実装段階において点滅はしない。また Tab キーや方向キーによるキャレットの移動ができ, 左方向キー (もしくは Tab キー), 右方向キー (もしくは Shift+Tab キー) による左右対称な移動を達成している。右方向キーを断続的に押下した場合におけるキャレットの表示, 及びその移動の様子を図 2 に示す。左方向キーを断続的に押下した場合には図 2 の逆順を辿る。

表 1: Padovani らの指標への対応状況

指標	対応	備考
入力スロットの表示	○	－
編集位置の表示とその移動	△	左右移動のみ可
編集領域の選択	○	－
カット（コピー）＆ペースト	×	実装予定
テンプレートとオーバーレイ	○	－
数式の直接編集	△	四則演算などのみ可

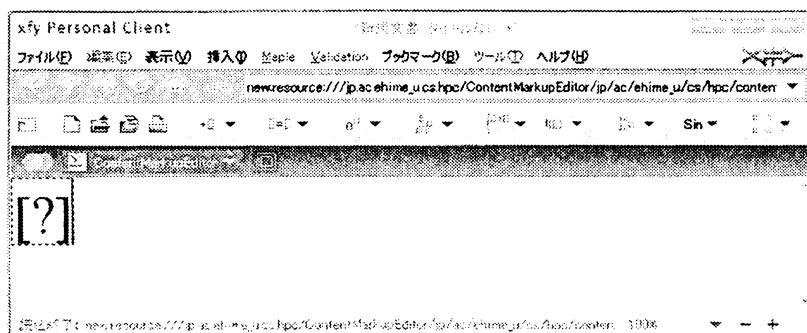


図 1: 入力スロット

**入力フォーカス** Content Markup プラグインでは、編集可能なフォーカス領域を入力フォーカスと呼び、入力フォーカスの状態により編集できる内容を変化させている。入力フォーカスの状態を図 3 に示す。それぞれの状態が示す内容は以下の通りである。

- FOCUS\_SELECTED 状態

破線の矩形で囲まれて示され、その領域にある数式が編集可能な状態を示す。この状態にあり、かつキャレットが左側（右側）にある場合に DELETE（BACKSPACE）キーが入力されると、FOCUS\_DELETABLE 状態に遷移する。

- FOCUS\_CONTEXT 状態

背景色が色付けされて示され、その領域にある数式が編集可能な状態を示す。この状態は ci 要素や cn 要素等のトークン要素が持つテキストノード値を編集するために用いる。

- FOCUS\_DELETABLE 状態

文字色や背景色が反転されて示され、その領域にある数式が削除可能な状態を示す。この状態にあり、かつキャレットが左側（右側）にある場合に DELETE（BACKSPACE）キーが入力されると数式が削除される。

- FOCUS\_OUT 状態

他の状態のように視覚的な変化はなく、その領域が編集対象でない状態を示す。

Content Markup プラグインではテンプレートによる入力方式を採用する。総和を入力する様子を図 4、及び図 5 に示す。

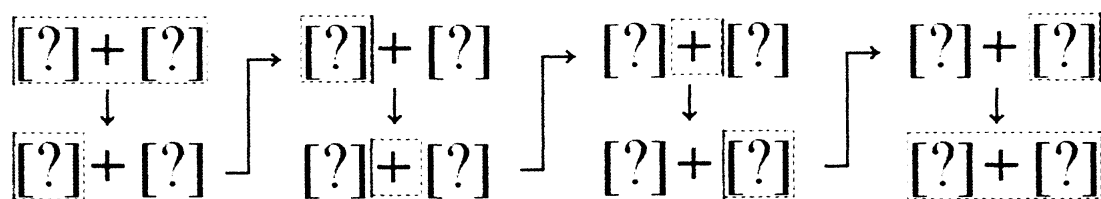


図 2: キャレットの表示と移動

<b>[?]</b>	FOCUS_SELECTED状態
[?]	FOCUS_CONTEXT状態
<b>[?]</b>	FOCUS_DELETABLE状態
[?]	FOCUS_OUT状態

図 3: 入力フォーカスの状態

また Content Markup プラグインでは Integre MathML Equation Editor と同様にキーボードからのオーバーレイ入力をサポートする。このオーバーレイ入力により、特に四則演算や多項式では中置記法を意識した自然な数式の入力を行うことができる。すなわち、数式  $x + y - 1$  を入力する場合には “x, +, y, -, 1” の順にキーを入力すればよい。

### 3 エディタの評価

本研究では Padovani らの示す指標に基づき、xly 上で Content Markup を編集するためのプラグインの開発を行った。しかしながら、Content Markup プラグインのユーザビリティが既存のエディタよりも優れているか否かについて言及することはできない。本節では Content Markup プラグインの入力性能の向上

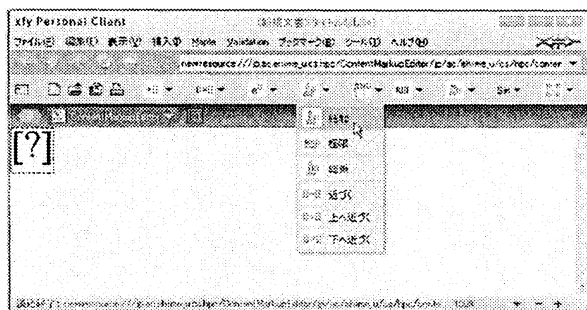


図 4: テンプレートの選択

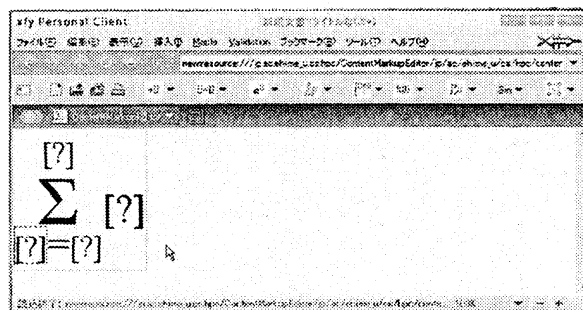


図 5: テンプレート入力

を目指すため、Content Markup プラグインのユーザビリティ評価を行い、その評価結果からより良い入力インタフェースについて検討する。

評価対象とするエディタは、Content Markup プラグイン、Formulator 3.7 MathML Weaver、Integre MathML Equation Editor の3つである。

エディタのユーザビリティを評価する一般的な手法として、テストユーザを用いる方法がある。テストユーザを用いる方法ではユーザビリティを評価するための課題が遂行される。

本研究ではテストユーザ（学生7名）に数式の入力課題（全10題）を与え、各エディタによる数式の入力コストを計測・集計する。テスト時には、ユーザに対し事前に操作方法の説明を行い、基本的な利用方法を理解してもらった。本研究で用いた数式の入力課題は表2の通りである。

表 2: ユーザビリティ評価に用いた数式一覧

課題番号	数式
1	$x + y \times z$
2	$(x + y) \times z$
3	$2x^2 - 4x + 4 = 2(x - 2)^2$
4	$\sum_{k=0}^{\infty} (k^2 + k + 1)$
5	$\lim_{n \rightarrow \infty} \sum_{k=0}^n k^2$
6	$\int_0^1 (x + 1) dx$
7	$\int_{x \in C} f(x) dx$
8	$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
9	$h(x) = \begin{cases} -x & x \leq -1 \\ \sin(x) & -1 < x < 1 \\ x & x \geq 1 \end{cases}$
10	$-e^{\log_2(1-t)}$

ここで、入力コストを、ある一つの数式課題を入力し終えるまでにかかったキーのタイプ数とマウスのクリック数の和と定義する。また  $i$  番目の課題についての入力コストを  $C_i$  で表し、入力課題すべての入力コストの総和を  $C_{ALL}$  と表す。また、 $i$  番目の数式に対する平均入力コストは  $C_{AV_i} = \sum_{i=1}^7 \frac{C_i}{7}$  である。

得られた入力コストから各エディタのユーザビリティを評価するために、仮説検定による評価を行う。ここで、仮説検定にはテストユーザ数の規模から鑑みてスチューデントの  $t$  検定（以下、 $t$  検定）を用いる。本研究では左片側検定問題として扱い、Content Markup プラグインの平均入力コストは既存のエディタの平均入力コストよりも小さいと言えるかどうかを評価する。この問題における帰無仮説、及び対立仮説を以下に示す。

**[帰無仮説]** Content Markup プラグインの平均入力コストは既存エディタと比べて差はない

**[対立仮説]** Content Markup プラグインの平均入力コストは既存エディタと比べて小さい

また有意水準 0.05 として検定を行う。

この仮説検定を各エディタ間において総入力コスト、および各数式課題の入力コストについて行う。また検定のためのツールとして統計解析システムである R を用いた。

### 3.1 計測結果

テストユーザ用いて得られた平均入力コストの計測結果を表 3 に示す。ただし、各値は小数点第二位を四捨五入したものである。

表 3: 各課題に対する平均入力コスト

	Content Markup プラグイン	Formulator 3.7 MathML Weaver	Integre MathML Equation Editor
課題 1	24	105.4	10.6
課題 2	24	105.4	18.3
課題 3	162	247	44.4
課題 4	47.7	53.1	42.3
課題 5	31.6	41	29.7
課題 6	22.3	27.6	24.4
課題 7	52.1	46.6	34.6
課題 8	44.3	33	66.4
課題 9	102.6	80	131.6
課題 10	37	23	43.9
全課題	549.4	682	446.1

Content Markup プラグインと各既存のエディタの間で t 検定による左片側検定を行った結果得られた  $p$  値を表 4 に示す。表 4 より、すべての課題について  $p$  値が有意水準 0.05 を下回ることではなく、帰無仮説を棄却できる有意な差は得られなかった。

逆に、Formulator 3.7 MathML Weaver における課題 8 や課題 10 ではその  $p$  値は 0.95 を上回った。すなわち、これは Formulator 3.7 MathML Weaver に対して Content Markup プラグインの平均入力コストが有意水準 0.05 で大きいと言えることを示している。また、Integre MathML Equation Editor における課題 1 や課題 3 に関しても同様に言える。

したがって、 $p$  値が 0.95 未満であるすべての課題について、Content Markup プラグインの平均入力コストは有意水準 0.05 で既存のエディタよりも小さくなるとは言えない。すなわち、Content Markup プラグインと既存のエディタとの間に平均入力コストに差はないと結論付けることができる。

課題 1 や課題 3 は四則演算や多項式等の数式入力課題である。Integre MathML Equation Editor は入力された数式を内部で構文解析することで Content Markup を出力するため、その入力インタフェースはコマンドライン入力と同様に行うことができる。そのため、テンプレート入力である Content Markup プラグインの平均入力コストの方が大きいという結果の原因になったと考えられる。

課題 8 や課題 10 は行列等の入力課題であり、上下左右へのキャレットの移動により編集を容易にできる。しかしながら、Content Markup プラグインはキャレットの上下の移動に関して未実装である。このため、Formulator 3.7 MathML Weaver に対して Content Markup プラグインの平均入力コストが大きいという

表 4: t 検定により得られた  $p$  値

	Formulator 3.7 MathML Weaver	Integre MathML Equation Editor
課題 1	0.1904	0.9845
課題 2	0.1689	0.8223
課題 3	0.1731	0.9963
課題 4	0.3505	0.7109
課題 5	0.159	0.629
課題 6	0.1251	0.3402
課題 7	0.6786	0.88
課題 8	0.9842	0.06509
課題 9	0.8824	0.1302
課題 10	0.9619	0.3815
全課題	0.2377	0.8684

結果が得られたと考えられる。同様な入力課題として課題 9 がある。この課題では  $p$  値が 0.8824 を示しており、Content Markup プラグインよりも平均入力コストが小さいという傾向が見られる。

一方、上下左右へのキャレットの移動が可能な Integre MathML Equation Editor に対しては有意水準 0.10 で Content Markup プラグインの平均入力コストの方が小さいと言えるという結論が得られた。これは Integre MathML Equation Editor の行列の入力方法が、行の追加は **Enter** キー、列の追加はカンマキーのみで行うという特殊な方法であり、ユーザを混乱させたためと考えられる。

以上から、今後の課題として数式のコマンドライン入力インタフェースの導入や、キャレットの上下移動への対応等が挙げられる。

## 4 まとめ

本研究では、数式の数学的意味を記述する Content Markup の作成・編集を目的とし、Content Markup の数学的意味の保持やその利用について検討・提案した。我々の Content Markup エディタは従来の Content Markup エディタと比較して次の新しい特徴を持っている。

- xfy のコンポーネントとして実装されているので、他のコンポーネントと連携しながら、数式を含む様々な文書の編集に利用できる。
- 編集の補助ツールとして数式処理システムの機能の一部が利用できる（実装および評価予定）。

これらの特徴を生かして、数式の入力インタフェースの更なる改善を課題としている。

本研究では、既存の Content Markup エディタとの間で入力コストによるユーザビリティ評価を行った。その結果、Content Markup エディタについて入力インタフェースの改善を図る必要性があることが分かった。

また、一般に Presentation Markup の入力手法である直接入力テンプレート入力よりユーザビリティは高いということが知られている。そこで、以下の課題についても取り組んでいくことを考えている。

- Presentation Markup から Content Markup に変換する方法の検討



## 参 考 文 献

- [1] Luca Padovani and Ricardo Solmi: An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics, MKM 2004, Lecture Notes in Computer Science Vol.3119, pp.302-316, 2004
- [2] XML, <http://www.w3.org/TR/xml/>
- [3] XHTML, <http://www.w3.org/TR/xhtml1/>
- [4] SVG, <http://www.w3.org/TR/SVG11/>
- [5] MathML, <http://www.w3.org/TR/MathML2/>
- [6] xfy, <http://www.xfy.com/>
- [7] Formulator 3.7 MathML Weaver, <http://mmlsoft.com/projects/formulator/>
- [8] Integre MathML Equation Editor, <http://www.integretechpub.com/zed/>
- [9] MathML Test Suite, <http://www.w3.org/Math/testsuite/>